

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-358757

(43)Date of publication of application : 26.12.2001

(51)Int.Cl.

H04L 12/56

(21)Application number : 2000-176775

(71)Applicant : ATR ADAPTIVE COMMUNICATIONS RES LAB

(22)Date of filing : 13.06.2000

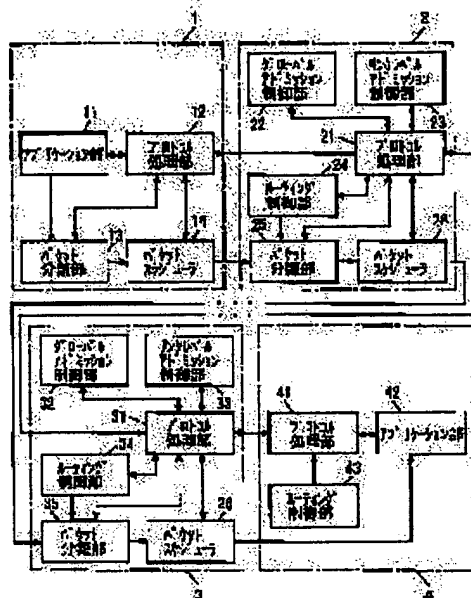
(72)Inventor : TANEDA KAZUMASA

(54) NODE DEVICE, DATA TRANSFER SYSTEM AND DATA TRANSFERRING METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a node device capable of improving throughput by eliminating deadlocks without being affected by user actions and also always and fairly allocating resources, and a data transfer system and a data transfer method which use the node device.

SOLUTION: When a transmitting host terminal device 1 transfers a packet to a receiving host terminal device 4, a protocol processing part 31 of the node device 3 connected to the terminal device 4 transmits a reservation message for reserving resources in place of the terminal device 4 according to global admission control performed by a global admission control part 32 when the terminal device 4 requests the resources and also transmits a reservation cancel message for canceling a reserved state in the case of receiving a reservation error message for the transmitted reservation message.



LEGAL STATUS

[Date of request for examination] 21.03.2001

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration] dismissal

[Date of final disposal for application] 27.05.2003

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

BEST AVAILABLE COPY

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2001-358757 ✓
(P2001-358757A)

(43) 公開日 平成13年12月26日 (2001. 12. 26)

(51) Int.Cl.⁷
H 0 4 L 12/56

識別記号

F I
H 0 4 L 11/20

テーマコード(参考)
1 0 2 C 5 K 0 3 0

審査請求 有 請求項の数13 O L (全 20 頁)

(21) 出願番号 特願2000-176775(P2000-176775)

(22) 出願日 平成12年6月13日 (2000. 6. 13)

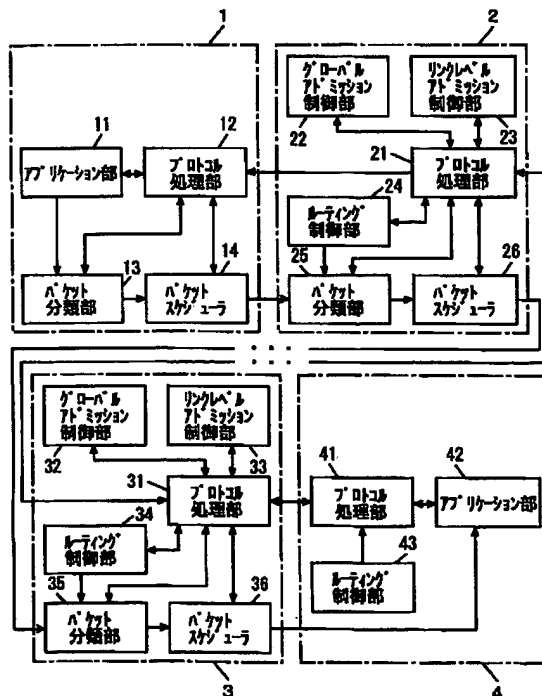
(71) 出願人 396011680
株式会社エイ・ティ・アール環境適応通信
研究所
京都府相楽郡精華町光台二丁目2番地2
(72) 発明者 種田 和正
京都府相楽郡精華町光台二丁目2番地2
株式会社エイ・ティ・アール環境適応通信
研究所内
(74) 代理人 100098305
弁理士 福島 祥人 (外1名)
Fターム(参考) 5K030 GA03 HA08 JA11 JT02 LC01
LC09 LD18

(54) 【発明の名称】 ノード装置、データ転送システムおよびデータ転送方法

(57) 【要約】

【課題】 ユーザの行為に影響されることなく、デッドロックを解消してスループットを向上することができるとともに、常に公正にリソースを割り当てることができるノード装置、このノード装置を用いたデータ転送システムおよびデータ転送方法を提供する。

【解決手段】 送信側ホスト端末装置1から受信側ホスト端末装置4へパケットを転送する際、受信側ホスト端末装置4に接続されたノード装置3のプロトコル処理部31が、グローバルアドミSSION制御部32によるグローバルアドミSSION制御に従い、受信側ホスト端末装置4からリソースを要求された場合に、受信側ホスト端末装置4に代わってリソースを予約するための予約メッセージを送信するとともに、送信された予約メッセージに対する予約エラーメッセージを受信した場合に予約状態を取り消すための予約取り消しメッセージを送信する。



【特許請求の範囲】

【請求項 1】 ネットワークを介して送信側端末装置から受信側端末装置へ転送されるデータの中継するノード装置であって、

前記ノード装置が前記受信側端末装置に接続され、前記受信側端末装置からリソースを要求された場合、前記受信側端末装置に代わってリソースを予約するための予約メッセージを送信する予約メッセージ送信手段と、前記予約メッセージ送信手段から送信された予約メッセージに対する予約エラーメッセージを受信した場合に予約状態を取り消すための予約取り消しメッセージを送信する予約取り消しメッセージ送信手段とを備えることを特徴とするノード装置。

【請求項 2】 前記予約メッセージ送信手段は、前記予約取り消しメッセージ送信手段がアドミッション制御の失敗による予約エラーメッセージを受信した場合に予約取り消しメッセージを送信した後、予約メッセージの送信から所定時間経過後に予約メッセージを再送することを特徴とする請求項 1 記載のノード装置。

【請求項 3】 前記予約メッセージ送信手段は、送信した予約メッセージに対する応答メッセージを所定期間受信しなかった場合に予約メッセージを再送することを特徴とする請求項 1 または 2 記載のノード装置。

【請求項 4】 前記予約メッセージ送信手段は、複数のセッションがリソース予約を待っている場合に、古いセッションの予約メッセージを優先的に送信することを特徴とする請求項 1～3 のいずれかに記載のノード装置。

【請求項 5】 複数のセッションがリソース予約を待っている場合に、リソース予約を待っているセッションの中から最も古いセッションを抽出し、抽出された最も古いセッションにリソースを優先的に割り当てるリソース割り当て手段をさらに備えることを特徴とする請求項 1～4 のいずれかに記載のノード装置。

【請求項 6】 複数のセッションがリソース予約を待っている場合に、重いセッションにリソースを優先的に割り当てるリソース割り当て手段をさらに備えることを特徴とする請求項 1～4 のいずれかに記載のノード装置。

【請求項 7】 複数のセッションがリソース予約を待っている場合に、リソース予約を待っているセッションの中から最も古いセッションを抽出し、抽出された最も古いセッションにリソースを優先的に割り当てるとともに、残りのリソースの中から重いセッションにリソースを優先的に割り当てるリソース割り当て手段をさらに備えることを特徴とする請求項 1～4 のいずれかに記載のノード装置。

【請求項 8】 前記リソース割り当て手段は、セッションの重みとしてセッションが必要とする帯域幅とホップ数との乗算値を用い、前記乗算値に応じてセッションを複数のグループに分割し、前記乗算値およびセッションの競合数に応じて各グループごとにリソースを割り当て

ることを特徴とする請求項 6 または 7 記載のノード装置。

【請求項 9】 複数のノード装置をリンクさせたネットワークを介して送信側端末装置から受信側端末装置へデータを転送するデータ転送システムであって、

前記複数のノード装置は、前記受信側端末装置に接続される出口側ノード装置を含み、

前記出口側ノード装置は、請求項 1～4 のいずれかに記載のノード装置を含むことを特徴とするデータ転送システム。

【請求項 10】 前記複数のノード装置の各々は、複数のセッションがリソース予約を待っている場合に、リソース予約を待っているセッションの中から最も古いセッションを抽出し、抽出された最も古いセッションにリソースを優先的に割り当てるリソース割り当て手段を備えることを特徴とする請求項 9 記載のデータ転送システム。

【請求項 11】 前記複数のノード装置の各々は、複数のセッションがリソース予約を待っている場合に、重いセッションにリソースを優先的に割り当てるリソース割り当て手段を備えることを特徴とする請求項 9 記載のデータ転送システム。

【請求項 12】 前記複数のノード装置の各々は、複数のセッションがリソース予約を待っている場合に、リソース予約を待っているセッションの中から最も古いセッションを抽出し、抽出された最も古いセッションにリソースを優先的に割り当てるとともに、残りのリソースの中から重いセッションにリソースを優先的に割り当てるリソース割り当て手段を備えることを特徴とする請求項 9 記載のデータ転送システム。

【請求項 13】 複数のノード装置をリンクさせたネットワークを介して送信側端末装置から受信側端末装置へデータを転送するデータ転送方法であって、前記受信側端末装置がリソースを要求した場合、前記受信側端末装置に接続される出口側ノード装置が前記受信側端末装置に代わってリソースを予約するための予約メッセージを送信するステップと、前記出口側ノード装置が前記予約メッセージに対する予約エラーメッセージを受信した場合に予約状態を取り消すための予約取り消しメッセージを送信するステップとを含むことを特徴とするデータ転送方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ネットワークを介して送信側端末装置から受信側端末装置へ転送されるデータの中継するノード装置、このノード装置を用いたデータ転送システムおよびデータ転送方法に関するものである。

【0002】

【従来の技術】統合サービスパケット交換ネットワーク

(ISP N)における保証サービスでは、ネットワークユーザからの予約要求に対してネットワークのリソース(伝送帯域幅)を排他的に割り当てるためにリソース予約プロトコル(RSVP: Resource Reservation Protocol)およびアドミッション制御を用い、遅延を許容しないアプリケーションに対してエンドツーエンドで一定の転送時間を保証している。

【0003】ここで、リソース予約プロトコルは、ネットワークを構成するノード装置およびホスト端末装置にセッションの予約状態を設定するためのプロトコルであり、セッションは、ユーザ間の一方方向のデータフローである。また、アドミッション制御は、現在の残りの伝送帯域幅を考慮して到達した各予約要求に対してリソースを予約すべきか否かを決定するものである。

【0004】図12は、セッションを2点間で一方方向にセットアップするのに成功した場合のリソース予約プロトコルの手順を示す図である。

【0005】図12に示すように、パスメッセージ(Path message)が送信側ホスト端末装置から出力されると、セッションに対するリソースの予約が開始される。パスメッセージは、各ノード装置にパス状態を設定し、予約メッセージ(Resv message)が伝送される経路を規定するために使用される。

【0006】パスメッセージが受信側ホスト端末装置に到達した後、必要とされる伝送帯域幅が決定され、予約メッセージが送信される。予約メッセージは、パスメッセージが通過した経路の各ノード装置に予約状態を設定し、全てのリンクに例えばr個のリソースを予約する。

【0007】予約メッセージが送信側ホスト端末装置に到達すると、予約が成功し、セッションが開始される。セッションのパケットは、セッションに予約されたリソースを用いることにより送信側ホスト端末装置から受信側ホスト端末装置へ送信される。パケットの送信完了後、パス取り消しメッセージ(PathTear message)および予約取り消しメッセージ(ResvTear message)が送信され、セッションに予約された全てのリソースが解放される。

【0008】ここで、各ノード装置は、隣接する上流側のノード装置または送信側ホスト端末装置からセッションのパスメッセージを周期的に受信すると、セッションのパス状態を維持するとともに、隣接する下流側のノード装置または受信側ホスト端末装置からセッションの予約メッセージを周期的に受信すると、セッションの予約状態を維持する。

【0009】一方、各ノード装置は、セッションの各状態のライフタイム内に当該セッションに対するメッセージを受信しなければ、その状態を取り消し、このような機構をソフト状態機構(soft-state mechanism)といい、以下の説明では、メッセージに対応する状態が存在する場合に「リフレッシュ」メッセージの語を使用し、

メッセージに対応する状態が存在しない場合に「ニュー」メッセージの語を使用する。

【0010】予約状態のライフタイム T_{life} は、リフレッシュ期間 T_{fresh} に基づき各ノード装置により計算され、各予約メッセージに含まれている。リフレッシュ期間 T_{fresh} が小さい場合、ライフタイム T_{life} は通常小さい値をとる。受信側ホスト端末装置がリフレッシュ予約メッセージの送信を停止した場合、 T_{fresh} が小さいと、予約されたリソースのライフタイムが短くなる。

【0011】図13は、セッションを2点間で一方方向にセットアップするのに失敗した場合のリソース予約プロトコルの手順を示す図である。

【0012】図13に示すように、要求を満足するのに十分なリソースを持たないすなわち予約されていないリソースの数がr個より少ないリンク装置に予約メッセージが遭遇すると、アドミッション制御に失敗する。このとき、予約エラーメッセージ(ResvError message)が送信され、アドミッション制御が失敗したことが受信側ホスト端末装置に通知される。受信側ホスト端末装置は、予約メッセージにより予約した全てのリソースを解放する場合、予約エラーメッセージを受信した後に、予約取り消しメッセージを送信する。

【0013】

【発明が解決しようとする課題】上記のようにして、リソースが予約されるが、予約されているが使用されていないリソースは暫定的予約状態にある。例えば、セッションのセットアップの長期間の遅延(セッションの全てのリソースの予約に使用される時間)、もしくはアドミッション制御の失敗の後または送信側ホスト端末装置/受信側ホスト端末装置がセッションを終了した後のリソースの解放の長期間の遅延により、リソースが暫定的予約状態になる。暫定的予約状態にあるリソースは、スループットを低下させる。

【0014】リソースは、暫定的予約状態にあるときに無駄に使用され、暫定的予約状態にあるリソースは、より多くの暫定的予約状態のリソースを生み出す。例えば、2人のユーザが同時にリソースを要求し、一方の要求に対して予約されたリソースが解放されたときにだけ他方の要求が満足され、一方のユーザが予約したリソースを解放しない場合に、デッドロックが発生する。

【0015】また、n個のセッションに対してn個のリソース r_1, \dots, r_n が個別に予約され、リソース r_1 がリソース r_2 により保持されているリソースを要求し、リソース r_2 がリソース r_3 により保持されているリソースを要求し、以降同様にリソース r_n がリソース r_1 により保持されているリソースを要求する場合、n個のリソース r_1, \dots, r_n が循環待ち状態になり、以下の2つの条件を満たす場合にデッドロックが発生する。

【0016】(1) 受信側ホスト端末が予約取り消しメ

ッセージを送信しない。

(2) $T_{ret} < T_{life}$

ここで、 T_{ret} は受信側ホスト端末装置の予約再送間隔であり、予約再送間隔 T_{ret} がライフタイム T_{life} より小さい場合、暫定的予約状態にあるリソース r_1, \dots, r_n のライフタイム T_{life} が満了しなくなる。

【0017】上記のデッドロックによるスループットの低下を防止するため、以下の2つの方法が提案されている。第1の方法は、全ての受信側ホスト端末装置が予約エラーメッセージを受信する度に予約取り消しメッセージを送信し、暫定的予約状態を直接的に取り消す方法であり、この方法を以下「ET (explicit tear down) 法」と呼ぶ。第2の方法は、全ての受信側ホスト端末装置がライフタイムが終了するまで予約メッセージの送信を停止し、暫定的予約状態を間接的に取り消す方法であり、この方法を以下「IT (implicit tear down) 法」と呼ぶ。

【0018】IT法における予約再送方法の一つは、指数的に予約再送間隔を長くしていく方法であり、全ての受信側ホスト端末装置は、予約エラーメッセージを受信する度に予約再送間隔 T_{ret} を倍にする。この結果、予約が失敗する度に予約再送間隔 T_{ret} が指数的に増大され、最終的には、デッドロックの条件(2)が満足されなくなる。

【0019】上記のET法およびIT法は、ネットワークユーザが全て協調的であるという状態を前提とするものであり、全てのユーザが2つの方法のうちの1つを受け入れることが必要になる。しかしながら、協調的でないユーザが何回も予約を繰り返したり、または、ユーザのホスト端末装置が正常に動作しない場合があり、実際にはすべてのユーザが常に協調的であるとは限らず、従来のリソース予約プロトコルでは、このようなユーザの行為に対してデッドロックを避けることができない。

【0020】また、アドミッション制御では、内部のリソースの利用状況を確認してリソースに余裕がある場合に予約要求に応じてセッションにリソースを割り当てる。すなわち、予約されていないリソースの数が要求リソースの数以上の場合にだけ、リソースが割り当てられる。このアドミッション制御による方法は、利用状況に基づく方法と呼ばれ、セッションを確立するためにリソースを公正に割り当てることが本質的に困難である。ここで、公正なリソースの割り当てとは、最初の予約メッセージがネットワークに到達した順番にセッションが確立されるようなリソースの割り当てをいうものとする。

【0021】上記のアドミッション制御のように利用状況に基づきリソースを割り当てする場合、以下の2つの点において公正にリソースを割り当てることができない。

【0022】第1に、多数のリソースを要求するセッション(重いセッション)のセットアップ遅延が大きくなる傾向にある。セッションの重さは、セッションを確立

するために要求されるリソースの全体数であり、要求される帯域幅をR、リソースが予約される経路のホップ数(ノード数)をhとすると、 Rh により表される。ここで、帯域幅Rまたはホップ数hが増加すると、予約の成功率が減少し、重いセッションのセットアップは軽いセッションのセットアップより遅くなる。

【0023】第2に、長時間確立されるまで待っているセッション(古いセッション)は、リソースを予約することができない。セッションの経過時間は、セッションをセットアップするための待ち時間を表し、具体的には、ネットワークにおけるセッションの最初の予約メッセージの到達時から現在までの期間である。利用状況に基づく方法は、競合するセッションの経過時間を考慮しないため、セッションが同じリンクのリソースを予約するために互いに競合し、リソースがまだ予約されていない場合、競合する新しいセッションにリソースが割り当てられることがあり、リソースを公正に割り当てることができない。

【0024】本発明の目的は、ユーザの行為に影響されることなく、デッドロックを解消してスループットを向上することができるとともに、常に公正にリソースを割り当てることができるノード装置、このノード装置を用いたデータ転送システムおよびデータ転送方法を提供することである。

【0025】

【課題を解決するための手段および発明の効果】(1)

第1の発明

第1の発明に係るノード装置は、ネットワークを介して送信側端末装置から受信側端末装置へ転送されるデータを中継するノード装置であって、ノード装置が受信側端末装置に接続され、受信側端末装置からリソースを要求された場合、受信側端末装置に代わってリソースを予約するための予約メッセージを送信する予約メッセージ送信手段と、予約メッセージ送信手段から送信された予約メッセージに対する予約エラーメッセージを受信した場合に予約状態を取り消すための予約取り消しメッセージを送信する予約取り消しメッセージ送信手段とを備えるものである。

【0026】第1の発明に係るノード装置においては、ノード装置が受信側端末装置に接続され、受信側端末装置からリソースを要求された場合、受信側端末装置に代わってリソースを予約するための予約メッセージを送信するとともに、予約メッセージに対する予約エラーメッセージを受信した場合に予約状態を取り消すための予約取り消しメッセージを送信している。したがって、ユーザの行為すなわち受信側端末装置の動作に関わらず、ノード装置が受信側端末装置に代わって予約メッセージおよび予約エラーメッセージを送信し、公正かつ協調的に予約状態の要求および取り消しを行うことができる。この結果、ユーザの行為に影響されることなく、デッドロ

7

ックを解消してスループットを向上することができるとともに、常に公正にリソースを割り当てることができる。

【0027】(2) 第2の発明

第2の発明に係るノード装置は、第1の発明に係るノード装置の構成において、予約メッセージ送信手段は、予約取り消しメッセージ送信手段がアドミッション制御の失敗による予約エラーメッセージを受信した場合に予約取り消しメッセージを送信した後、予約メッセージの送信から所定時間経過後に予約メッセージを再送するものである。

【0028】この場合、アドミッション制御の失敗による予約エラーメッセージを受信した場合に予約取り消しメッセージを送信した後、予約メッセージの送信から所定時間経過後に予約メッセージを再送しているため、アドミッション制御の失敗によりリソースを予約できなかった場合でも、予約状態を一旦取り消した後に、公正かつ協調的に予約メッセージを再送することができる。

【0029】(3) 第3の発明

第3の発明に係るノード装置は、第1または第2の発明に係るノード装置の構成において、予約メッセージ送信手段は、送信した予約メッセージに対する応答メッセージを所定期間受信しなかった場合に予約メッセージを再送するものである。

【0030】この場合、予約メッセージに対する応答メッセージを所定期間受信しなかった場合に予約メッセージを再送しているため、メッセージの消失等によりリソースを予約できなかった場合でも、公正かつ協調的に予約メッセージを再送することができる。

【0031】(4) 第4の発明

第4の発明に係るノード装置は、第1～第3のいずれかの発明に係るノード装置の構成において、予約メッセージ送信手段は、複数のセッションがリソース予約を待っている場合に、古いセッションの予約メッセージを優先的に送信するものである。

【0032】この場合、複数のセッションがリソース予約を待っている場合に、古いセッションの予約メッセージを優先的に送信しているため、古いセッションから順にセッションのセットアップを行うことができる。

【0033】(5) 第5の発明

第5の発明に係るノード装置は、第1～第4のいずれかの発明に係るノード装置の構成において、複数のセッションがリソース予約を待っている場合に、リソース予約を待っているセッションの中から最も古いセッションを抽出し、抽出された最も古いセッションにリソースを優先的に割り当てるリソース割り当て手段をさらに備えるものである。

【0034】この場合、複数のセッションがリソース予約を待っている場合に、リソース予約を待っているセッションの中から最も古いセッションを抽出し、抽出され

8

た最も古いセッションにリソースを優先的に割り当てているので、最初の予約メッセージがネットワークに到達した順番にセッションを確立することができる。

【0035】(6) 第6の発明

第6の発明に係るノード装置は、第1～第4のいずれかの発明に係るノード装置の構成において、複数のセッションがリソース予約を待っている場合に、重いセッションにリソースを優先的に割り当てるリソース割り当て手段をさらに備えるものである。

【0036】この場合、複数のセッションがリソース予約を待っている場合に、重いセッションにリソースを優先的に割り当てているので、重いセッションに対しても公正にリソースを割り当てることができる。

【0037】(7) 第7の発明

第7の発明に係るノード装置は、第1～第4のいずれかの発明に係るノード装置の構成において、複数のセッションがリソース予約を待っている場合に、リソース予約を待っているセッションの中から最も古いセッションを抽出し、抽出された最も古いセッションにリソースを優先的に割り当てるとともに、残りのリソースの中から重いセッションにリソースを優先的に割り当てるリソース割り当て手段をさらに備えるものである。

【0038】この場合、複数のセッションがリソース予約を待っている場合に、リソース予約を待っているセッションの中から最も古いセッションを抽出し、抽出された最も古いセッションにリソースを優先的に割り当てるとともに、残りのリソースの中から重いセッションにリソースを優先的に割り当てているので、古いセッションを最優先してリソースを割り当てることができるとともに、重いセッションにも優先的にリソースを割り当てることができる。

【0039】(8) 第8の発明

第8の発明に係るノード装置は、第6または第7の発明に係るノード装置の構成において、リソース割り当て手段は、セッションの重みとしてセッションが必要とする帯域幅とホップ数との乗算値を用い、乗算値に応じてセッションを複数のグループに分割し、乗算値およびセッションの競合数に応じて各グループごとにリソースを割り当てるものである。

【0040】この場合、セッションの重みとしてセッションが必要とする帯域幅とホップ数との乗算値を用い、乗算値に応じてセッションを複数のグループに分割し、乗算値およびセッションの競合数に応じて各グループごとにリソースを割り当てているので、セッションの重みおよび競合数に対して適応的にリソースを割り当てることができる。

【0041】(9) 第9の発明

第9の発明に係るデータ転送システムは、複数のノード装置をリンクさせたネットワークを介して送信側端末装置から受信側端末装置へデータを転送するデータ転送シ

システムであって、複数のノード装置は、受信側端末装置に接続される出口側ノード装置を含み、出口側ノード装置は、第1～第4のいずれかの発明に係るノード装置を含むものである。

【0042】第9の発明に係るデータ転送システムにおいては、ノード装置が受信側端末装置に接続され、受信側端末装置からリソースを要求された場合、受信側端末装置に代わってリソースを予約するための予約メッセージを送信するとともに、予約メッセージに対する予約エラーメッセージを受信した場合に予約状態を取り消すための予約取り消しメッセージを送信している。したがって、ユーザの行為すなわち受信側端末装置の動作に関わらず、ノード装置が受信側端末装置に代わって予約メッセージおよび予約エラーメッセージを送信し、公正かつ協調的に予約状態の要求および取り消しを行うことができる。この結果、ユーザの行為に影響されることなく、デッドロックを解消してスループットを向上することができる。とともに、常に公正にリソースを割り当てることができる。

【0043】(10) 第10の発明

第10の発明に係るデータ転送システムは、第9の発明に係るデータ転送システムの構成において、複数のノード装置の各々は、複数のセッションがリソース予約を待っている場合に、リソース予約を待っているセッションの中から最も古いセッションを抽出し、抽出された最も古いセッションにリソースを優先的に割り当てるリソース割り当て手段を備えるものである。

【0044】この場合、複数のセッションがリソース予約を待っている場合に、リソース予約を待っているセッションの中から最も古いセッションを抽出し、抽出された最も古いセッションにリソースを優先的に割り当てているので、最初の予約メッセージがネットワークに到達した順番にセッションを確立することができる。

【0045】(11) 第11の発明

第11の発明に係るデータ転送システムは、第9の発明に係るデータ転送システムの構成において、複数のノード装置の各々は、複数のセッションがリソース予約を待っている場合に、重いセッションにリソースを優先的に割り当てるリソース割り当て手段を備えるものである。

【0046】この場合、複数のセッションがリソース予約を待っている場合に、重いセッションにリソースを優先的に割り当てているので、重いセッションに対しても公正にリソースを割り当てることができる。

【0047】(12) 第12の発明

第12の発明に係るデータ転送システムは、第9の発明に係るデータ転送システムの構成において、複数のノード装置の各々は、複数のセッションがリソース予約を待っている場合に、リソース予約を待っているセッションの中から最も古いセッションを抽出し、抽出された最も古いセッションにリソースを優先的に割り当てるととも

に、残りのリソースの中から重いセッションにリソースを優先的に割り当てるリソース割り当て手段を備えるものである。

【0048】この場合、複数のセッションがリソース予約を待っている場合に、リソース予約を待っているセッションの中から最も古いセッションを抽出し、抽出された最も古いセッションにリソースを優先的に割り当てるとともに、残りのリソースの中から重いセッションにリソースを優先的に割り当てているので、古いセッションを最優先してリソースを割り当てることができる。とともに、重いセッションにも優先的にリソースを割り当てることができる。

【0049】(13) 第13の発明

第13の発明に係るデータ転送方法は、複数のノード装置をリンクさせたネットワークを介して送信側端末装置から受信側端末装置へデータを転送するデータ転送方法であって、受信側端末装置がリソースを要求した場合、受信側端末装置に接続される出口側ノード装置が受信側端末装置に代わってリソースを予約するための予約メッセージを送信するステップと、出口側ノード装置が予約メッセージに対する予約エラーメッセージを受信した場合に予約状態を取り消すための予約取り消しメッセージを送信するステップとを含むものである。

【0050】第13の発明に係るデータ転送方法においては、ノード装置が受信側端末装置に接続され、受信側端末装置からリソースを要求された場合、受信側端末装置に代わってリソースを予約するための予約メッセージを送信するとともに、予約メッセージに対する予約エラーメッセージを受信した場合に予約状態を取り消すための予約取り消しメッセージを送信している。したがって、ユーザの行為すなわち受信側端末装置の動作に関わらず、ノード装置が受信側端末装置に代わって予約メッセージおよび予約エラーメッセージを送信し、公正かつ協調的に予約状態の要求および取り消しを行うことができる。この結果、ユーザの行為に影響されることなく、デッドロックを解消してスループットを向上することができる。とともに、常に公正にリソースを割り当てることができる。

【0051】

【発明の実施の形態】以下、本発明の一実施の形態のノード装置を用いたデータ転送システムについて図面を参照しながら説明する。図1は、本発明の一実施の形態のノード装置を用いたデータ転送システムの構成を示すブロック図である。

【0052】図1において、データ転送システムは、送信側ホスト端末装置1、ノード装置2、3および受信側ホスト端末装置4を備える。ノード装置2は、送信側ホスト端末装置1に接続され、入口側のノード装置となり、ノード装置3は、受信側ホスト端末装置4に接続され、出口側のノード装置となる。ノード装置2とノード

装置3との間には、複数のノード装置がデータ伝送路となるリンクを介して接続され、ネットワークを形成しているが、説明を容易にするために図示を省略している。なお、ネットワークとしては、例えば、イントラネットが用いられるが、この例に特に限定されず、インターネット等の他のネットワークを用いてもよい。

【0053】送信側ホスト端末装置1、ノード装置2、3および受信側ホスト端末装置4は、従来のリソース予約プロトコルを後述するグローバルアドミッション制御およびリンクレベルアドミッション制御により拡張した拡張リソース予約プロトコルに従い、公正かつ協調的にリソースの割り当てを行い、例えば、送信側ホスト端末装置1からノード装置2およびノード装置3を介して受信側ホスト端末装置4へデータ（パケット）が転送される。なお、送信側ホスト端末装置1、ノード装置2、3および受信側ホスト端末装置4におけるグローバルアドミッション制御およびリンクレベルアドミッション制御による動作以外の動作は、従来のリソース予約プロトコルに従い、アドミッション制御およびポリシー制御等により所定の動作が行われる。

【0054】送信側ホスト端末装置1は、サーバまたはパーソナルコンピュータ等から構成され、アプリケーション部11、プロトコル処理部12、パケット分類部13およびパケットスケジューラ14を含み、所定のソフトウェアまたはハードウェアによりこれらの各ブロックの機能が実現される。

【0055】アプリケーション部11は、所定のアプリケーションを実行し、当該アプリケーションが必要とするリソースの予約をプロトコル処理部12に要求するとともに、リソースの予約が完了した後に各パケットをパケット分類部13へ出力する。

【0056】プロトコル処理部12は、アプリケーション部11からのリソースの予約要求を受け、リソース予約プロトコルに従い、パスメッセージ、予約メッセージ等の種々のRSVPメッセージをノード装置2のプロトコル処理部21と送受信するとともに、パケット分類部13およびパケットスケジューラ14に所定のパラメータを設定する。

【0057】パケット分類部13は、アプリケーション部11から出力される各パケットのセッション等を決定し、パケットスケジューラ14へ各パケットを出力する。パケットスケジューラ14は、各セッションに対して予約されたリソース数に応じて各パケットの転送スケジュールを管理し、各パケットを所定の順序で転送する。

【0058】ノード装置2は、ルータ等から構成され、プロトコル処理部21、グローバルアドミッション制御部22、リンクレベルアドミッション制御部23、ルーティング制御部24、パケット分類部25およびパケットスケジューラ26を含み、所定のソフトウェアまたは

ハードウェアによりこれらの各ブロックの機能が実現される。

【0059】グローバルアドミッション制御部22は、後述するグローバルアドミッション制御を行うようにプロトコル処理部21を制御する。リンクレベルアドミッション制御部23は、後述するリンクレベルアドミッション制御を行うようにプロトコル処理部21を制御する。なお、グローバルアドミッション制御は、ノード装置が出口側のノード装置となる場合に使用されるため、ノード装置2が出口側のノード装置とならない場合にはグローバルアドミッション制御部22を省略してもよい。

【0060】ルーティング制御部24は、内部にパス状態を有し、RSVPメッセージおよびパケットを転送する経路をルーティングし、ルーティングされた経路に沿ってRSVPメッセージおよびパケットが転送されるようにプロトコル処理部21およびパケット分類部24を制御する。

【0061】プロトコル処理部21は、リンクレベルアドミッション制御部23によるリンクレベルアドミッション制御を含む拡張リソース予約プロトコルに従い、ルーティング制御部24によりルーティングされた経路に沿って種々のRSVPメッセージを送受信するとともに、接続されているリンクのリソースの状態を管理し、パケット分類部25およびパケットスケジューラ26に所定の各パラメータを設定する。

【0062】パケット分類部25は、パケットスケジューラ14から出力される各パケットのセッション等を決定し、パケットスケジューラ26へ各パケットを出力する。パケットスケジューラ26は、各セッションに対して予約されたリソース数に応じて各パケットの転送スケジュールを管理し、各パケットを所定の順序で所定のリンクへ転送する。

【0063】ノード装置3は、ルータ等から構成され、プロトコル処理部31、グローバルアドミッション制御部32、リンクレベルアドミッション制御部33、ルーティング制御部34、パケット分類部35およびパケットスケジューラ36を含み、所定のソフトウェアまたはハードウェアによりこれらの各ブロックの機能が実現される。

【0064】グローバルアドミッション制御部32は、グローバルアドミッション制御を行うようにプロトコル処理部31を制御する。リンクレベルアドミッション制御部33は、リンクレベルアドミッション制御を行うようにプロトコル処理部31を制御する。

【0065】ルーティング制御部34は、内部にパス状態を有し、RSVPメッセージおよびパケットを転送する経路をルーティングし、ルーティングされた経路に沿ってRSVPメッセージおよびパケットが転送されるようにプロトコル処理部31およびパケット分類部35を

10

20

30

40

50

制御する。

【0066】プロトコル処理部31は、グローバルアドミッション制御部32によるグローバルアドミッション制御およびリンクレベルアドミッション制御部33によるリンクレベルアドミッション制御を含む拡張リソース予約プロトコルに従い、ルーティング制御部34によりルーティングされた経路に沿って種々のRSVPメッセージを送受信するとともに、接続されているリンクのリソースの状態を管理し、パケット分類部35およびパケットスケジューラ36に所定の各パラメータを設定する。

【0067】パケット分類部35は、パケットスケジューラ26から出力される各パケットのセッション等を決定し、パケットスケジューラ36へ各パケットを出力する。パケットスケジューラ36は、各セッションに対して予約されたリソース数に応じて各パケットの転送スケジュールを管理し、各パケットを所定の順序で所定のリンクへ転送する。

【0068】受信側ホスト端末装置4は、サーバまたはパーソナルコンピュータ等から構成され、プロトコル処理部41、アプリケーション部42およびルーティング制御部43を含み、所定のソフトウェアまたはハードウェアによりこれらの各ブロックの機能が実現される。

【0069】プロトコル処理部41は、リソース予約プロトコルに従い、ノード装置3のプロトコル処理部33と種々のRSVPメッセージを送受信する。アプリケーション部42は、所定のアプリケーションを実行し、パケットスケジューラ36から出力される各パケットを受信する。ルーティング制御部43は、RSVPメッセージ等を転送する経路をルーティングし、ルーティングされた経路に沿ってRSVPメッセージ等が転送されるようにプロトコル処理部41を制御する。

【0070】本実施の形態では、送信側ホスト端末装置1が送信側端末装置に相当し、受信側ホスト端末装置4が受信側端末装置に相当し、ノード装置2、3がノード装置に相当し、グローバルアドミッション制御部32およびプロトコル処理部31が予約メッセージ送信手段および予約取り消しメッセージ送信手段に相当する。また、リンクレベルアドミッション制御部23、33およびプロトコル処理部21、31がリソース割り当て手段に相当し、ノード装置3が出口側ノード装置に相当する。

【0071】次に、ノード装置3のグローバルアドミッション制御部32によるグローバルアドミッション制御について詳細に説明する。

【0072】ノード装置3は、リソースを要求する受信側ホスト端末装置4に直接接続され、セッションのセットアップ期間中に、プロトコル処理部33は、グローバルアドミッション制御部32によるグローバルアドミッション制御に従い、受信側ホスト端末装置4の要求を終

端し、受信側ホスト端末装置4の代わりに公正かつ協調的にリソースを要求する。

【0073】プロトコル処理部33は、公正にリソースを要求するために、継続しているセッションの経過時間を考慮してセッションのセットアップ順序を決定する。すなわち、プロトコル処理部33は、グローバルアドミッション制御部32によるグローバルアドミッション制御に従い、継続している古いセッションが確立されるまで継続している新しいセッションに対する予約メッセージの送信を停止する。

【0074】このとき、古いセッションに対するセットアップ遅延が長くなると、スループットが低下するため、予約メッセージの再送の数の増加に応じて予約の成功比率を増加させて古いセッションが早く確立されるように、リンクレベルアドミッション制御部23、33によりリンクレベルアドミッション制御が後述するように行われる。

【0075】また、協調的なリソースの要求を達成するために、プロトコル処理部33は、予約エラーメッセージを受信する度に予約取り消しメッセージを送信する。これにより暫定的予約状態が早急に取り消され、すばやい再送により循環待ち状態にある暫定的予約状態のリソースの増加を防止することができる。

【0076】例えば、2つのセッションに対する2つの暫定的予約状態のリソースr1、r2が循環待ち状態にある場合、予約取り消しメッセージによりリソースr1、r2に対する予約状態が取り消された後に、2つのセッションの予約メッセージが同時に再送されると、予約メッセージは高い確率で同じリソースr1、r2を予約し、この状態が長い間継続する。しかしながら、リンクレベルアドミッション制御部23、33によりリンクレベルアドミッション制御が後述するように行われ、再送の後により古いまたはより重いセッションがリソースを獲得することができるように制御される。

【0077】図2は、図1に示すデータ転送システムの拡張リソース予約プロトコルの手順を示す図である。

【0078】予約のセットアップ期間中のノード装置3の動作は、以下の3つの点で従来のリソース予約プロトコルによるメッセージ処理と異なる。

【0079】第1に、セッションの予約メッセージが受信側ホスト端末装置4からノード装置3に到達する度に、ノード装置3が予約状態になっていなければ、そのセッションのための予約状態が作られる。ここで、対応するセッションがノード装置3においてリソースを有していない予約状態をダミー予約状態というものとし、対応するセッションがまだ確立されていない予約状態をペンディング予約状態というものとする。このダミー予約状態は、常にペンディング予約状態となる。

【0080】ノード装置3は、継続している各セッションに対して、セッションが確立されるまでペンディング

10

20

30

40

50

予約状態を用いて予約メッセージを作成し、作成された予約メッセージを再送する。この場合、受信側ホスト端末装置4からみると、最初の予約メッセージが常にリソースの予約に成功し、次の予約メッセージはペンディング予約状態をリフレッシュするためにだけ使用されるようにみえる。

【0081】上記のようにして、グローバルアドミッション制御では、図2に示すように、ニュー予約メッセージの中継遅延時間 T_{relay} を決定し、セッションのセットアップの順序を決定する。中継遅延時間 T_{relay} は、

【0082】このように、複数のセッションがリソース予約を待っている場合に、古いセッションの予約メッセージを優先的に送信しているので、古いセッションから順にセッションのセットアップを行うことができる。

【0083】第2に、図2に示すように、上流側のノード装置、例えばノード装置2から予約エラーメッセージを受信したときにこのエラーがアドミッション制御の失敗によるものの場合、ノード装置3は、直ちに予約取り消しメッセージを送信し、以前の予約メッセージが送信された時刻から再送期間 T_{round} が経過した後、予約メッセージを再送する。エラーがアドミッション制御の失敗でない場合、ノード装置3は、受信側ホスト端末装置4へ予約エラーメッセージを中継するとともに、上流側のノード装置へ予約取り消しメッセージを送信し、さらに、ノード装置3の状態を取り消してセッションのセットアップをキャンセルする。

【0084】この場合、予約取り消しメッセージが送信された後に、予約メッセージの送信から再送期間 T_{round} 経過後に予約メッセージを再送しているの、アドミッション制御の失敗によりリソースを予約できなかった場合でも、予約状態を一旦取り消した後に、公正かつ協調的に予約メッセージを再送することができる。

【0085】第3に、図2に示すように、予約メッセージの送信後から再送期間 T_{loss} 内に予約エラーメッセージまたは予約確認メッセージ(ResvConf message)がノード装置3に到達しない場合、ノード装置3は予約メッセージを再送する。再送期間 T_{loss} は、効率を考慮すると、小さい値に設定されることが好ましいが、デッドロックを避けるため、再送期間 T_{loss} は、予約エラーメッセージが到達する前に予約メッセージが再送されないように十分大きい値にしなければならず、この範囲で適切な値に設定される。

【0086】この場合、予約メッセージに対する応答メッセージを再送期間 T_{loss} 内に受信しなかった場合に予約メッセージを再送しているの、メッセージの消失等によりリソースを予約できなかった場合でも、公正かつ協調的に予約メッセージを再送することができる。

【0087】なお、上記の予約確認メッセージは、送信側ホスト端末装置1に直接接続された入口側のノード装

置2により転送され、2つのノード間での予約の成功を出口側のノード装置3に知らせており、この予約確認メッセージは、リソース予約プロトコルの仕様に規定される予約確認メッセージと少し異なる。このため、全ての予約メッセージには、入口側のノード装置2のIPアドレスを含む。予約確認メッセージが到達した場合の予約状態を確立予約状態というものとする。

【0088】また、ノード装置3は、待機状態にある継続中のセッションのセットアップの順序を決定する際、例えば、 q -OSF(q oldest sessions first)法を用いる。この方法では、選択期間 T_s ごとに、継続中のセッションのグループの中から予約メッセージを送送する1つのセッションがラウンドロビンにより選択される。予約の成功(予約確認)または予約のキャンセル(予約エラー、予約取り消し、パス取り消しまたはライフタイムの満了)が発生する度に、グループ内のメンバーとなるセッションが更新され、常に q 個の最も古くから継続しているセッションからグループが構成される。この場合、再送期間 $T_{\text{round}} = q T_s$ となり、継続中のセッションの数 p が q 以上の場合、 $p = q$ に修正される。

【0089】また、デッドロックを避けるために転送が禁止されている場合、セッションの順番はスキップされ、上記のアドミッション制御の失敗による予約エラーメッセージの受信およびメッセージのロスタイマーの満了(T_{loss} の満了)の2つのイベントのうちの少なくとも1つが発生した後に予約メッセージが転送される。この場合、セッションがキャンセルされなければ、 $q = 1$ の場合、 q -OSF法は、最初に来たものが最初にサービスを受けるFCFS(first-come first-served)原理と等価となる。

【0090】上記のように、本実施の形態では、ノード装置3が受信側ホスト端末装置4に接続され、受信側ホスト端末装置4からリソースを要求された場合、グローバルアドミッション制御により、ノード装置3が受信側ホスト端末装置4に代わってリソースを予約するための予約メッセージを送信するとともに、予約メッセージに対する予約エラーメッセージを受信した場合に予約状態を取り消すための予約取り消しメッセージを送信している。

【0091】したがって、ユーザの行為すなわち受信側ホスト端末装置4の動作に関わらず、ノード装置3が予約メッセージおよび予約エラーメッセージを送信し、公正かつ協調的に予約状態の設定および取り消しを行うことができる。この結果、ユーザの行為に影響されことなく、デッドロックを解消することができるとともに常に公正にリソースを割り当てることができる。

【0092】次に、ノード装置2、3のリンクレベルアドミッション制御部23、33によるリンクレベルアドミッション制御について詳細に説明する。

【0093】ノード装置2およびノード装置3を含むネットワークを構成する複数のノード装置は、リンクレベルアドミッション制御部23、33等によるリンクレベルアドミッション制御に従い、現在の残りの伝送帯域幅だけでなく、要求しているセッションの特性、例えばセッションの重さ（要求しているリソースの数）またはセッションのライフタイム（セッションのセットアップの待ち時間）を考慮し、各予約要求に対してリソースを割り当てる。

【0094】すなわち、リンクレベルアドミッション制御部23、33によるリンクレベルアドミッション制御は、予約メッセージがノード装置に到達する度に重いおよび/または古いセッションがリソースを予約する確率が高くなるように、各予約要求（予約メッセージ）に対してリンクのリソースを割り当てる。本実施の形態では、q-OFS法とともに以下に説明する3種類のリンクレベルアドミッション制御方法のうちの一つを用いてリソースを割り当てている。

【0095】まず、第1のリンクレベルアドミッション制御方法として用いられる重み分類法（以下、WC法という）について説明する。

【0096】利用状況に基づく従来のアドミッション制御では、

$$r \geq R \quad (1)$$

の場合に、セッションヘリンクのリソースを割り当てる。ここで、Rは、セッションが要求する帯域幅であり、rは、リンクの予約されていない帯域幅である。

【0097】一方、WC法は、上記の(1)式を

$$r_i \geq R$$

に修正する。ここで、 r_i は重み分類iに属するセッションに割り当てることができる最大帯域幅を示し、例えば、 $i=1, 2, 3$ であり、 $i > j$ の場合、重み分類iのセッションは重み分類jのセッションより重くなる。また、セッションの重みを考慮して、全てのリンクに対して、 $r_1 \leq r_2 \leq r_3 = r$ とする。

【0098】WC法は、重み分類2のセッションおよび重み分類3のセッションの競合数により、 r_1 および r_2 を適応的に変化させる。例えば、両者の数が0の場合、 $r_1 = r_2 = r_3 = r$ となり、競合する重み分類3のセッションの数が増加した場合 r_1 および r_2 が減少され、重み分類3のセッションがリソースを予約する確率を高くする。

【0099】WC法では、プロトコル処理部21、31により予約メッセージの到達数をカウントすることにより、 r_1, r_2, r_3 を決定する。q-OFS法に従えば、1つのセッションの予約メッセージが、カウント期間Tc内に最大で $\lceil Tc/qTs \rceil$ 回（ここで、 $\lceil x \rceil$ はxの整数部を表す）リンクに到達する。したがって、リンク上のリソースの予約に失敗した重み分類iのセッションの予約メッセージの到達数がn $\lceil Tc/qTs \rceil$

+1以上の場合、リンクは少なくともn+1個の競合する重み分類iのセッションを有する。

【0100】ノード装置3のプロトコル処理部31は、セッションの重みに応じて各予約メッセージ内に重み分類iを挿入し、セッションの重みは、Rhにより計算される。ここで、Rは受信側ホスト端末装置4から送信される予約メッセージに含まれている要求帯域幅であり、hはバスメッセージに含まれている統合サービスをサポートするノード数である。

10 【0101】図3および図4は、WC法によるリンクレベルアドミッション制御方法を説明するためのフローチャートである。なお、図3および図4に示す $A_{i,R}$ は、リソースの予約に失敗した、重み分類iおよび要求帯域幅Rの予約メッセージの到達数のカウンタ値であり、Mは、ネットワークで利用可能な帯域幅である。

【0102】まず、図3に示すように、ステップS1において、ノード装置のプロトコル処理部は、帯域幅Rを要求し重み分類iに属するセッションsの予約メッセージ(s, R, i)を受信したか否かを判断し、受信した場合はステップS1へ移行し、受信していない場合は図4に示すステップS8へ移行する。

【0103】予約メッセージを受信した場合、ステップS2において、プロトコル処理部は、セッションsに対する予約状態が存在するか否かを判断し、予約状態が存在しない場合はステップS3へ移行し、予約状態が存在する場合は図4に示すステップS8へ移行する。

【0104】予約状態が存在しない場合、ステップS3において、プロトコル処理部は、 $r_i \geq R$ であるか否かを判断し、 $r_i \geq R$ の場合はステップS4へ移行し、 $r_i < R$ でない場合はステップS6へ移行する。

【0105】 $r_i \geq R$ の場合、ステップS4において、プロトコル処理部は、セッションsの要求帯域幅Rに対するリソースを予約する。次に、ステップS5において、プロトコル処理部は、 $k=1, 2, 3$ に対して、 $r_k - R$ を r_k に代入し、残りの各リソースを更新する。

【0106】一方、 $r_i < R$ でない場合、ステップS6において、プロトコル処理部は、 $i \geq 2$ であるか否かを判断し、 $i \geq 2$ の場合はステップS7へ移行し、 $i < 2$ でない場合は図4に示すステップS8へ移行する。 $i \geq 2$ の場合、ステップS7において、プロトコル処理部は、 $A_{i,R} + 1$ を $A_{i,R}$ に代入し、各カウンタ値を更新する。

【0107】次に、図4に示すように、ステップS8において、プロトコル処理部は、カウント期間Tcが満了したか否かを判断し、カウント期間Tcが満了した場合はステップS9へ移行し、ネットワークで利用可能な帯域幅Mに含まれる各要求帯域幅Rに対して以降の処理を行い、カウント期間Tcが満了していない場合はステップS1へ戻り、処理を繰り返す。

50 【0108】カウント期間Tcが満了した場合、ステッ

ブS9において、プロトコル処理部は、 $1 \leq A_{3,R} < 1 + [Tc/qTs]$ であるか否かを判断し、 $1 \leq A_{3,R} < 1 + [Tc/qTs]$ の場合はステップS10へ移行し、 $1 \leq A_{3,R} < 1 + [Tc/qTs]$ でない場合はステップS11へ移行する。 $1 \leq A_{3,R} < 1 + [Tc/qTs]$ の場合、ステップS10において、プロトコル処理部は、 $r_3 - R$ を r_2 に代入し、重み分類2のリソース r_2 を修正する。

【0109】一方、 $1 \leq A_{3,R} < 1 + [Tc/qTs]$ でない場合、ステップS11において、プロトコル処理部は、 $1 + [Tc/qTs] \leq A_{3,R}$ であるか否かを判断し、 $1 + [Tc/qTs] \leq A_{3,R}$ の場合はステップS12へ移行し、 $1 + [Tc/qTs] \leq A_{3,R}$ でない場合はステップS13へ移行する。 $1 + [Tc/qTs] \leq A_{3,R}$ の場合、重み分類3のセッションが競合しているので、ステップS12において、プロトコル処理部は、 $r_3 - 2R$ を r_2 に代入し、重み分類2のリソース r_2 がより小さくなるように修正する。

【0110】次に、ステップS13において、プロトコル処理部は、 $1 \leq A_{2,R} + A_{3,R} < 1 + [Tc/qTs]$ であるか否かを判断し、 $1 \leq A_{2,R} + A_{3,R} < 1 + [Tc/qTs]$ の場合はステップS14へ移行し、 $1 \leq A_{2,R} + A_{3,R} < 1 + [Tc/qTs]$ でない場合はステップS15へ移行する。 $1 \leq A_{2,R} + A_{3,R} < 1 + [Tc/qTs]$ の場合、ステップS14において、プロトコル処理部は、 $r_3 - R$ を r_1 に代入し、重み分類1のリソース r_1 を修正する。

【0111】一方、 $1 \leq A_{2,R} + A_{3,R} < 1 + [Tc/qTs]$ でない場合、ステップS15において、プロトコル処理部は、 $1 + [Tc/qTs] \leq A_{2,R} + A_{3,R}$ であるか否かを判断し、 $1 + [Tc/qTs] \leq A_{2,R} + A_{3,R}$ の場合はステップS16へ移行し、 $1 + [Tc/qTs] \leq A_{2,R} + A_{3,R}$ でない場合はステップS17へ移行する。 $1 + [Tc/qTs] \leq A_{2,R} + A_{3,R}$ の場合、重み分類2および重み分類3のセッションが競合しているので、ステップS16において、プロトコル処理部は、 $r_3 - 2R$ を r_1 に代入し、重み分類1のリソース r_1 がより小さくなるように修正する。

【0112】次に、ステップS17において、プロトコル処理部は、カウント値 $A_{2,R}$ 、 $A_{3,R}$ をリセットした後、ステップS1へ戻り、処理を繰り返す。

【0113】上記のWC法により、複数のセッションがリソース予約を待っている場合に、重いセッションにリソースを優先的に割り当てているので、重いセッションに対しても公正にリソースを割り当てることができる。また、セッションの重みとしてセッションが必要とする帯域幅 R とホップ数 h との乗算値を用い、 Rh に応じてセッションを複数のグループに分割し、 Rh およびセッションの競合数に応じて各グループごとにリソースを割り当てているので、セッションの重みおよび競合数に対

して適応的にリソースを割り当てることができる。

【0114】次に、第2のリンクレベルアドミッション制御方法として用いられる排他的保存法（以下、EP法という）について説明する。

【0115】EP法は、上記の（1）式を、

i) セッションが最も古い場合、 $r \geq R$

ii) その他の場合、 $r \geq R + R'$

のように修正することにより、同じリンクのリソースを要求するセッションの中から最も古い継続中のセッションに対してリソースを保存する。ここで、 R' は、最も古い継続中のセッションの要求帯域幅である。

【0116】ノード装置3から出力される各予約メッセージは、受信側ホスト端末装置4からノード装置3へ最初に予約メッセージが到着した時間を示すタイムスタンプ t を含む。タイムスタンプ t は、セッションのセットアップ期間内にだけ運ばれ、タイムスタンプ t を含むセッション s の予約メッセージは、セッション s が継続中のセッションであることを示す。

【0117】EP法では、上流側のノード装置において、プロトコル処理部によりカウント期間 Tc に到達した予約メッセージのタイムスタンプを比較することにより、各リンクに対する最も古い継続中のセッションが検出され、次のカウント期間 Tc において継続中の最も古いセッションが要求する帯域幅が排他的に保存される。すなわち、EP法では、ネットワークにおいて、最も古いセッションが確立されるまで、このセッションのためのリソースが保存される。

【0118】図5は、EP法によるリンクレベルアドミッション制御方法を説明するためのフローチャートである。

【0119】図5に示すように、まず、ステップS21において、ノード装置のプロトコル処理部は、帯域幅 R を要求しタイムスタンプ t を有するセッション s の予約メッセージ (s, R, t) を受信したか否かを判断し、受信した場合はステップS22へ移行し、受信していない場合はステップS28へ移行する。

【0120】予約メッセージを受信した場合、ステップS22において、プロトコル処理部は、 $t_{old} > t$ であるか否かを判断し、 $t_{old} > t$ の場合はステップS23へ移行し、 $t_{old} > t$ でない場合はステップS24へ移行する。

【0121】 $t_{old} > t$ の場合、ステップS23において、プロトコル処理部は、 t を t_{old} に代入し、最も古いタイムスタンプ t_{old} を更新するとともに、 R を R_{old} に代入し、最も古いタイムスタンプ t_{old} を有するセッション s の要求帯域幅 R_{old} を更新する。

【0122】次に、ステップS24において、プロトコル処理部は、セッション s に対する予約状態が存在するか否かを判断し、予約状態が存在しない場合はステップS25へ移行し、予約状態が存在する場合はステップS

28へ移行する。

【0123】予約状態が存在しない場合、ステップS25において、プロトコル処理部は、 $r \geq R$ かつ $t = t'$ 、または $r \geq R + R'$ であるか否かを判断し、 $r \geq R$ かつ $t = t'$ 、または $r \geq R + R'$ の場合はステップS26へ移行し、 $r \geq R$ かつ $t = t'$ 、または $r \geq R + R'$ でない場合はステップS28へ移行する。

【0124】 $r \geq R$ かつ $t = t'$ 、または $r \geq R + R'$ の場合、ステップS26において、プロトコル処理部は、セッションsの要求帯域幅Rに対するリソースを予約する。次に、ステップS27において、プロトコル

処理部は、 $r - R$ を r に代入し、残りのリソース r を更新する。

【0125】次に、ステップS28において、プロトコル処理部は、カウント期間Tcが満了したか否かを判断し、カウント期間Tcが満了した場合はステップS29へ移行し、カウント期間Tcが満了していない場合はステップS1へ戻り、処理を繰り返す。

【0126】カウント期間Tcが満了した場合、ステップS29において、プロトコル処理部は、 t_{old} を t' に代入し、最も古いタイムスタンプ t_{old} を t' に格納するとともに、 R_{old} を R に代入し、最も古いタイムスタンプ t_{old} を有するセッションsの要求帯域幅 R_{old} を R' に格納し、また、 t_{old} に初期値 ∞ を代入するとともに、 R_{old} に初期値0を代入した後、ステップS1へ戻り、処理を繰り返す。

【0127】上記のEP法により、複数のセッションがリソース予約を待っている場合に、リソース予約を待っているセッションの中から最も古いセッションが抽出され、抽出された最も古いセッションにリソースが優先的に割り当てられているので、最初の予約メッセージがネットワークに到達した順番にセッションを確立することができる。

【0128】次に、第3のリンクレベルアドミッション制御方法として用いられるWC法およびEP法を組み合わせた方法（以下、C法という）について説明する。

【0129】C法は、WC法とEP法とを組み合わせたものであり、上記の（1）式が以下のように修正される。

【0130】i) セッションが最も古い場合、 $r \geq R$
ii) その他の場合、 $r_j \geq R + R'$

上記の条件により、C法でも、EP法と同様に、同じリンクのリソースを要求するセッションの中から最も古い継続中のセッションに対してリソースが保存される。また、C法は、予約されていない残りのリソースをWC法により割り当てる。

【0131】図6は、C法によるリンクレベルアドミッション制御方法を説明するためのフローチャートである。なお、C法のカウント期間Tcの満了時の処理は、WC法およびEP法のカウント期間Tcの満了時の処理

を単に連結したものであるため、詳細な説明を省略するが、当該処理が以下に説明するステップS38、S40の次に挿入され、上記と同様に処理される。

【0132】図6に示すように、まず、ステップS31において、ノード装置のプロトコル処理部は、帯域幅Rを要求しタイムスタンプ t を有し重み分類 i に属するセッションsの予約メッセージ(s, R, i, t)を受信したか否かを判断し、受信した場合はステップS32へ移行し、受信していない場合はカウント期間Tcの満了時の処理終了後ステップS31へ戻り、以降の処理を継続する。

【0133】予約メッセージを受信した場合、ステップS32において、プロトコル処理部は、 $t_{old} > t$ であるか否かを判断し、 $t_{old} > t$ の場合はステップS33へ移行し、 $t_{old} > t$ でない場合はステップS34へ移行する。

【0134】 $t_{old} > t$ の場合、ステップS33において、プロトコル処理部は、 t を t_{old} に代入し、最も古いタイムスタンプ t_{old} を更新するとともに、 R を R_{old} に代入し、最も古いタイムスタンプ t_{old} を有するセッションsの要求帯域幅 R_{old} を更新する。

【0135】次に、ステップS34において、プロトコル処理部は、セッションsに対する予約状態が存在するか否かを判断し、予約状態が存在しない場合はステップS35へ移行し、予約状態が存在する場合はカウント期間Tcの満了時の処理終了後ステップS31へ戻り、以降の処理を継続する。

【0136】予約状態が存在しない場合、ステップS35において、プロトコル処理部は、 $r \geq R$ かつ $t = t'$ 、または $r_j \geq R + R'$ であるか否かを判断し、 $r \geq R$ かつ $t = t'$ 、または $r_j \geq R + R'$ の場合はステップS36へ移行し、 $r \geq R$ かつ $t = t'$ 、または $r_j \geq R + R'$ でない場合はステップS39へ移行する。

【0137】 $r \geq R$ かつ $t = t'$ 、または $r_j \geq R + R'$ の場合、ステップS36において、プロトコル処理部は、セッションsの要求帯域幅Rに対するリソースを予約する。次に、ステップS37において、プロトコル処理部は、 $r - R$ を r に代入し、残りのリソース r を更新する。次に、ステップS38において、プロトコル処理部は、 $k = 1, 2, 3$ に対して、 $r_k - R$ を r_k に代入し、残りの各リソースを更新した後、カウント期間Tcの満了時の処理終了後ステップS31へ戻り、処理を繰り返す。

【0138】一方、 $r \geq R$ かつ $t = t'$ 、または $r_j \geq R + R'$ でない場合、ステップS39において、プロトコル処理部は、 $i \geq 2$ であるか否かを判断し、 $i \geq 2$ の場合はステップS40へ移行し、 $i \geq 2$ でない場合はカウント期間Tcの満了時の処理終了後ステップS31へ戻り、処理を繰り返す。 $i \geq 2$ の場合、ステップS40において、プロトコル処理部は、 $A_{i,R} + 1$ を $A_{i,R}$ に

代入し、各カウント値を更新した後、さらにカウント期間 T_c の満了時の処理終了後ステップ S 31 へ戻り、処理を繰り返す。

【0139】上記の C 法により、複数のセッションがリソース予約を待っている場合に、リソース予約を待っているセッションの中から最も古いセッションを抽出し、抽出された最も古いセッションにリソースを優先的に割り当てるとともに、残りのリソースの中から重いセッションにリソースを優先的に割り当てているので、古いセッションを最優先してリソースを割り当てることができるとともに、重いセッションにも優先的にリソースを割り当てることができる。

【0140】また、セッションの重みとしてセッションが必要とする帯域幅 R とホップ数 h との乗算値 Rh を用い、 Rh に応じてセッションを複数のグループに分割し、 Rh およびセッションの競合数に応じて各グループごとにリソースを割り当てているので、セッションの重みおよび競合数に対して適応的にリソースを割り当てることができる。

【0141】次に、上記のように構成されたデータ転送システムを用いて 3 種類のネットワークモデルを作成し、各ネットワークモデルの性能をシミュレーションした結果について説明する。

【0142】図 7 は、シミュレーションに用いた 3 種類のネットワークモデルのトポロジーを示す図である。なお、図中の実線はリンクを示し、全てのリンクは、双方向、完全二重送であり、同じ伝送帯域幅（1.5 Mビット/秒）を有し、図中の黒丸がノード（例えば、ノード装置 2、3 に相当）を示している。

【0143】図 7 の (a) は、第 1 のネットワークモデルを示し、ノード数は 14 個であり、ノード 1 個当たりのリンク数の平均値、最小値および最大値はそれぞれ 3.0、2、4 であり、リンクの伝搬遅延の平均値、最小値および最大値はそれぞれ 9.24、4、20 である。

【0144】図 7 の (b) は、第 2 のネットワークモデルを示し、ノード数は 28 個であり、ノード 1 個当たりのリンク数の平均値、最小値および最大値はそれぞれ 3.0、2、4 であり、リンクの伝搬遅延の平均値、最小値および最大値はそれぞれ 9.21、4、14 である。

【0145】図 7 の (c) は、第 3 のネットワークモデルを示し、ノード数は 42 個であり、ノード 1 個当たりのリンク数の平均値、最小値および最大値はそれぞれ 3.0、2、4 であり、リンクの伝搬遅延の平均値、最小値および最大値はそれぞれ 9.25、4、14 である。

【0146】第 1～第 3 のネットワークモデルは、ノードの数によりネットワークの大きさを表すため、ノード 1 個当たりのリンク数の平均値およびリンクの伝送遅延

の平均値をほぼ同じ値にしている。

【0147】各ノードは、バケット用の入力バッファと、リソース予約プロトコルおよびルーティングプロトコルに使用されるメッセージの伝送用の入力バッファとの 2 つの入力バッファを有し、各入力バッファのバケットおよびメッセージは、最初に来たものが最初に要求を満たすように処理され、出力バッファの 1 つに割り当てられ、転送される。各出力バッファ内のバケットおよびメッセージの転送順序は、重みに応じて公正な待ち行列を作るアルゴリズム (weight fair queueing algorithm) によってスケジューリングされている。なお、シミュレーションでは、全ての入力バッファおよび出力バッファの容量は無限であり、バケットによるメッセージは決して消失しないものと仮定している。

【0148】各リンクの転送能力は、12 個に均等分割され、1 つがベストエフォートサービスに割り当てられ、他の 1 つがプロトコルを伝送するために割り当てられ、残りが必要に応じて各セッションに割り当てられる。したがって、各リンクは、10 個のリソースを有する。

【0149】メッセージを伝送するための処理時間は 2 ms であり、バケットの処理時間は 0 ms に設定される。選択期間 T_s およびカウント期間 T_c はそれぞれ、20 ms、200 ms であり、 q -OSF 法の q は 4 個である。

【0150】また、WC 法を用いた場合の重み分類は、第 1 のネットワークモデルでは、 $Rh \leq 1$ を重み分類 1 とし、 $2 \leq Rh \leq 6$ を重み分類 2 とし、 $9 \leq Rh$ を重み分類 3 とし、第 2 のネットワークモデルでは、 $Rh \leq 2$ を重み分類 1 とし、 $3 \leq Rh \leq 6$ を重み分類 2 とし、 $9 \leq Rh$ を重み分類 3 とし、第 3 のネットワークモデルでは、 $Rh \leq 3$ を重み分類 1 とし、 $4 \leq Rh \leq 9$ を重み分類 2 とし、 $12 \leq Rh$ を重み分類 3 とした。

【0151】この結果、第 1 のネットワークモデルでは、重み分類 1 のセッションが 10 % となり、重み分類 2 のセッションが 50 % となり、重み分類 3 のセッションが 40 % となり、第 2 のネットワークモデルでは、重み分類 1 のセッションが 15 % となり、重み分類 2 のセッションが 50 % となり、重み分類 3 のセッションが 35 % となり、第 3 のネットワークモデルでは、重み分類 1 のセッションが 25 % となり、重み分類 2 のセッションが 45 % となり、重み分類 3 のセッションが 30 % となった。

【0152】次に、トラヒックソースについて説明する。セッションを確立するためのニューバスメッセージは、比率 λ (セッション/秒) のポアソン過程に従い、送信側ホスト端末装置から各ノード装置へ送信され、各ニューバスメッセージに対する出口側のノード装置がネットワーク全体で均一に選択される。受信側ホスト端末装置は、ニューバスメッセージの到着後すぐに予約メッ

セージを送出する。

【0153】各セッションは、1つのリソース ($R=1$) または3つのリソース ($R=3$) を同じ確率で要求し、 $M=\{1, 3\}$ となる。 $R=1$ および $R=3$ のセッションに対するパケットの生成比は、10パケット/秒および30パケット/秒であり、パケットのサイズは、平均サイズが512バイトで負の指数分布を有する。

【0154】各セッションのトラヒックは、ネットワーク内に注入される前にトークンパケットフィルタで成形される。トークンパケットサイズおよび最大パケットサイズは2048バイトであり、トークンパケットレートおよびピーク比はセッションの要求帯域幅 R と同一である。最大パケットサイズより大きなパケットは捨てられ、セッションに対するパケットの数は、 $(1, N_{max})$ の範囲で均一に分布している。ここで、 N_{max} は200、600、1800であり、セッションの平均ホールディング時間 T_{hold} は、 N_{max} の各値に対して、6.67秒、20秒、60秒である。ノードごとのベストエフォートパケットの到達比は、37パケット/秒である。

【0155】上記の各条件を前提として、シミュレーション時間 T_{sim} ($T_{sim}=90T_{hold}$) 内に確立されたセッションの総数 (スループット) および受信側ホスト端末装置からの最初の 10^4 個のニュー予約メッセージに対するセッションの最大セットアップ遅延 (最大遅延) に基づき、従来のIT法およびET法、ならびに本発明によるWC法、EP法およびC法について効率および公正さをシミュレーションにより評価した。なお、特に言及しなければ、全ての結果は同一のシミュレーションパラメータで5回のシミュレーションを行ったものの平均であり、WC法、EP法またはC法は、 $q-O SF$ 法とともに全てのノード装置に適用されている。

【0156】まず、各方法のスループットについて説明する。なお、シミュレーションの結果、ネットワークサイズはいずれの方法のスループットに対してもあまり影響せず、セッションのホールディング時間 T_{hold} のみがスループットに影響したため、ホールディング時間 T_{hold} の影響について以下に説明する。

【0157】図8は、ホールディング時間 T_{hold} が6.67秒のときの第1のネットワークモデルのスループット (確立されたセッション数) と負荷 (λT_{hold}) との関係を示す図であり、図9は、ホールディング時間 T_{hold} が60秒のときの第1のネットワークモデルのスループット (確立されたセッション数) と負荷 (λT_{hold}) との関係を示す図である。図中、黒菱形 (一点鎖線) はWC法を示し、白丸 (破線) はEP法を示し、黒四角 (二点鎖線) はC法を示し、 \times は $T_{ret}=1$ (秒) の場合のET法を示し、黒三角は $T_{ret}=2^n$ (秒) 場合 (初期再送間隔が1秒でその後の再送間隔が指数的に増加する場合) のET法を示し、黒逆三角は $T_{ret}=1$ (秒) および $T_{life}=0.9$ (秒) の場合のIT法を示し、黒丸は $T_{ret}=2^n$ (秒) および $T_{life}=60$ (秒) の場合のIT法を示している。

【0158】図8および図9に示すように、WC法、EP法およびC法は、ネットワークの大きさおよびセッションのホールディング時間によらず、安定したスループットを実現することができるとともに、IT法より高いスループットを実現することができることがわかった。

【0159】次に、負荷と、リソースの利用率および確立されたセッションの平均重さとの関係について説明する。リソースの利用率 (暫定的予約状態リソースを除く) は、第1のネットワークモデルを用いてホールディング時間 T_{hold} が6.67秒で各負荷 (λT_{hold}) が2.93、5.86、11.72の場合、 $T_{ret}=1$ (秒) のET法の場合に36.15 (%)、67.75 (%)、77.92 (%) となり、WC法の場合に33.77 (%)、61.08 (%)、62.03 (%) となり、EP法の場合に35.06 (%)、58.14 (%)、59.00 (%) となり、C法の場合に34.20 (%)、58.18 (%)、58.40 (%) となった。

【0160】また、確立されたセッションの平均重さは、第1のネットワークモデルを用いてホールディング時間 T_{hold} が6.67秒で各負荷 (λT_{hold}) が2.93、5.86、11.72の場合、 $T_{ret}=1$ (秒) のET法の場合に4.07、3.99、3.43となり、WC法の場合に3.89、3.82、3.84となり、EP法の場合に4.05、3.86、3.92となり、C法の場合に3.96、3.92、3.93となった。

【0161】上記のように、ET法では、負荷 (λT_{hold}) が増加すると、リソースの利用率は増加したが、確立されたセッションの平均重さは減少した。一方、WC法、EP法およびC法では、確立されたセッションの平均重さは、 λ に依存せず、ほぼ同じ値となった。なお、到達した全てのセッションが最初のパスメッセージにより確立される場合の平均重さは3.98である。

【0162】次に、各方法の公正さについて説明する。図10は、ホールディング時間 T_{hold} が20秒のときの第1のネットワークモデル (ネットワークサイズ (ノード数) $|S|=14$) の受信側ホスト端末装置からの最初の 10^4 個のニュー予約メッセージに対するセッションの最大セットアップ遅延 (最大遅延) と全到着率 ($\lambda |S|$) との関係を示す図であり、図11は、ホールディング時間 T_{hold} が20秒のときの第3のネットワークモデル (ネットワークサイズ $|S|=42$) の受信側ホスト端末装置からの最初の 10^4 個のニュー予約メッセージに対するセッションの最大セットアップ遅延 (最大遅延) と全到着率 ($\lambda |S|$) との関係を示す図である。図中、黒菱形 (一点鎖線) はWC法を示し、白丸 (破線) はEP法を示し、黒四角 (二点鎖線) はC法を示し、黒逆三角は $T_{ret}=1$ (秒) および $T_{life}=0.9$ (秒) の場合のIT法を示し、黒丸は $T_{ret}=2^n$ (秒) および $T_{life}=60$ (秒) の場合のIT法を示している。

示し、×は $T_{ret} = 1$ (秒) の場合の E T 法を示し、黒三角は $T_{ret} = 2^n$ (秒) 場合 (初期再送間隔が 1 秒でその後の再送間隔が指数的に増加する場合) の E T 法を示し、黒逆三角は $T_{ret} = 1$ (秒) および $T_{life} = 0.9$ (秒) の場合の I T 法を示し、黒丸は $T_{ret} = 2^n$ (秒) および $T_{life} = 60$ (秒) の場合の I T 法を示している。

【0163】図 10 および図 11 に示すように、従来の I T 法および E T 法は不安定な振る舞いを示すことがわかる。特に、指数的に再送間隔を長くしていく場合、臨

界値が小さくなる。一方、WC 法、E P 法および C 法は、全く安定な振る舞いを示すことがわかる。

【0164】また、WC 法、E P 法および C 法の振る舞いは、ネットワークサイズ S が小さければほぼ同じであるが、WC 法は、大きなネットワークに対しては不安定になる。これは、ネットワークサイズとともに 2 つのホ

スト端末装置間の平均ホップ数の増加によるものであり、ホップ数が増加すると、セッションのセットアップを失敗する比率が高くなるためである。

【0165】一方、E P 法および C 法は、大きなネット

ワークに対しても全く安定である。これは、セッションが確立されるまで、最も古い継続しているセッションのリソースをこれらの方法が保持するという特性によるものである。E P 法および C 法のただ 1 つの違いは、C 法は、ネットワークサイズが大きい場合、E P 法よりやや優れている点である。

【0166】また、 $\lambda |S| = 17$ (リクエスト/秒) の場合 (到着率が非常に高い場合) に 10 回シミュレーションを行った結果、最大遅延の最大値および標準偏差はそれぞれ、第 1 のネットワークモデルにおいて、WC

法では 3322, 266 となり、E P 法では 2187, 32 となり、C 法では 2188, 16 となり、第 2 のネットワークモデルにおいて、WC 法では 4725, 405 となり、E P 法では 1320, 14 となり、C 法では 1430, 48 となり、第 3 のネットワークモデルにおいて、WC 法では 34841, 3733 となり、E P 法では 2454, 171 となり、C 法では 2951, 465 となった。このように、E P 法および C 法の最大遅延の最大値は、第 1 ~ 第 3 のネットワークモデルで大きな差がなかった。

【0167】また、セッションのホールディング時間は最大遅延に直接影響するが、ホールディング時間 T_{hold} が 6.67 秒、20 秒、60 秒の各場合に対して最大遅延と負荷 (λT_{hold}) との間の関係は、 T_{hold} が 3 倍に増加すると最大遅延がほぼ同じ倍率で増大する点を除き、WC 法、E P 法および C 法に依存せず、全く同様であった。

【0168】上記の各シミュレーションの結果、WC

ず安定であった。また、WC 法、E P 法および C 法のセッションの最大セットアップ遅延は、全てのネットワークサイズ、全てのセッションのホールディング時間および全ての到着率において、従来の E T 法および I T 法より小さくなった。

【0169】このように、WC 法、E P 法および C 法は、非常に公正な振る舞いを実現することができるとともに、小さな最大遅延および安定なスループットを実現することができることがわかった。

【図面の簡単な説明】

【図 1】本発明の一実施の形態のノード装置を用いたデータ転送システムの構成を示すブロック図である。

【図 2】図 1 に示すデータ転送システムの拡張リソース予約プロトコルの手順を示す図である。

【図 3】WC 法によるリンクレベルアドミッション制御方法を説明するための第 1 のフローチャートである。

【図 4】WC 法によるリンクレベルアドミッション制御方法を説明するための第 2 のフローチャートである。

【図 5】E P 法によるリンクレベルアドミッション制御方法を説明するためのフローチャートである。

【図 6】C 法によるリンクレベルアドミッション制御方法を説明するためのフローチャートである。

【図 7】シミュレーションに用いた 3 種類のネットワークモデルのトポロジを示す図である。

【図 8】ホールディング時間が 6.67 秒のときの第 1 のネットワークモデルのスループットと負荷との関係を示す図である。

【図 9】ホールディング時間が 60 秒のときの第 1 のネットワークモデルのスループットと負荷との関係を示す図である。

【図 10】ホールディング時間が 20 秒のときの第 1 のネットワークモデルの最大遅延と全到着率との関係を示す図である。

【図 11】ホールディング時間が 60 秒のときの第 3 のネットワークモデルの最大遅延と全到着率との関係を示す図である。

【図 12】セッションを 2 点間で一方方向にセットアップするのに成功した場合のリソース予約プロトコルの手順を示す図である。

【図 13】セッションを 2 点間で一方方向にセットアップするのに失敗した場合のリソース予約プロトコルの手順を示す図である。

【符号の説明】

- 1 送信側ホスト端末装置
- 2, 3 ノード装置
- 4 受信側ホスト端末装置
- 11, 42 アプリケーション部
- 12, 21, 31, 41 プロトコル処理部
- 13, 25, 35 パケット分類部
- 14, 26, 36 パケットスケジューラ

29

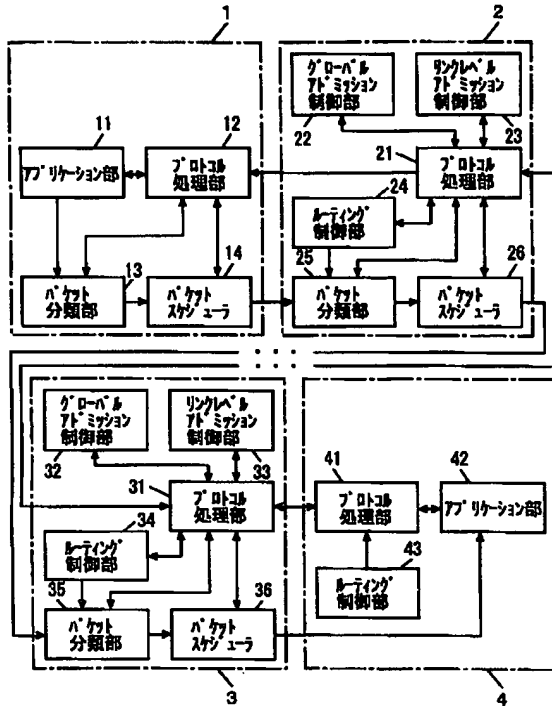
30

22, 32 グローバルアドミッション制御部

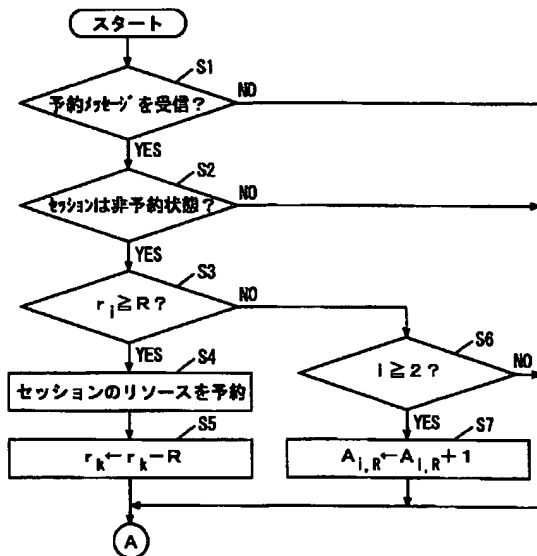
24, 34, 43 ルーティング制御部

23, 33 リンクレベルアドミッション制御部

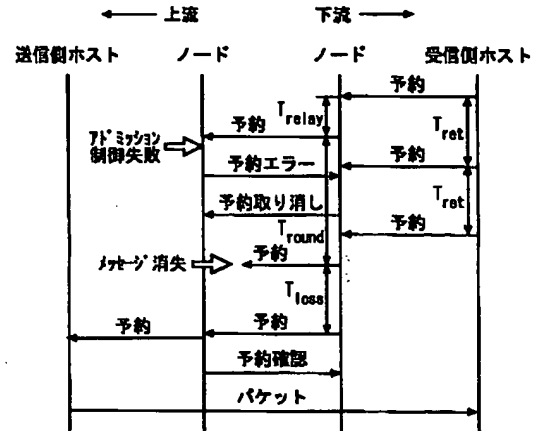
【図1】



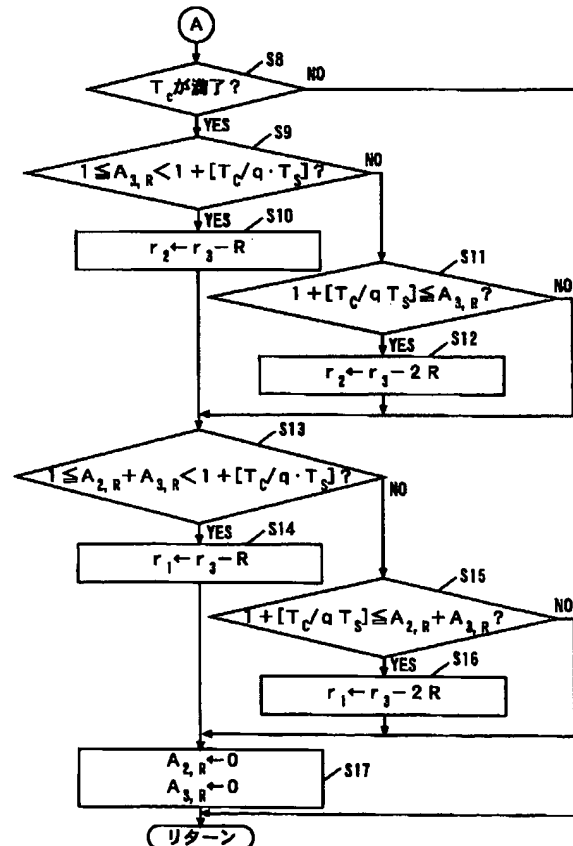
【図3】



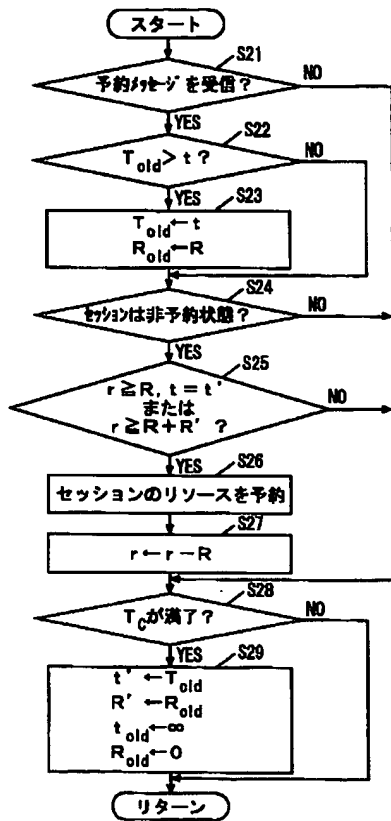
【図2】



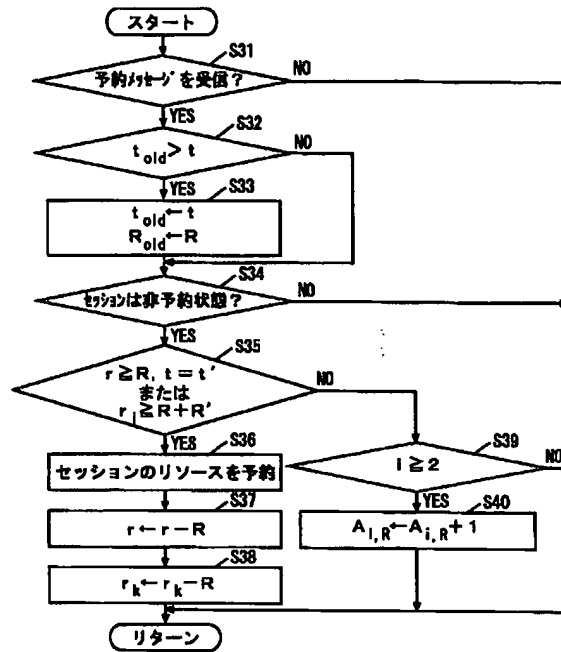
【図4】



【図5】

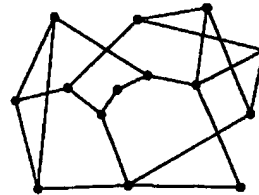


【図6】

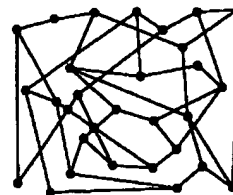


【図7】

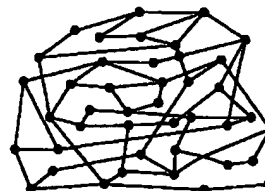
(a)



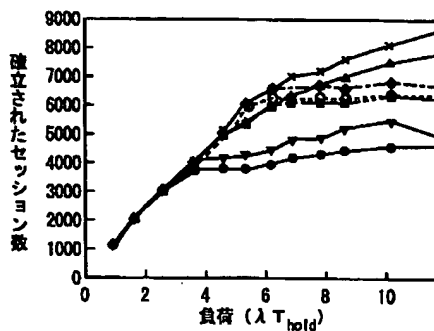
(b)



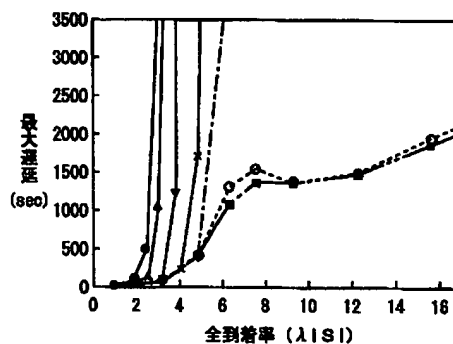
(c)



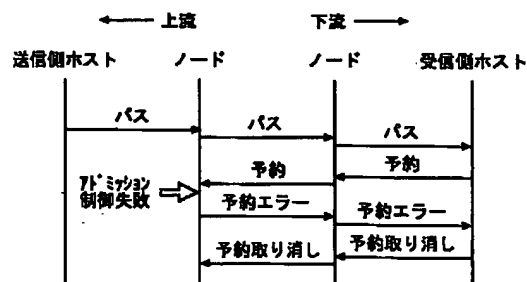
【図 9】



【图 1-1】



【図 13】



【補正内容】

【0073】プロトコル処理部33は、公正にリソースを要求するために、リソース予約を待っているセッションの経過時間を考慮してセッションのセットアップ順序を決定する。すなわち、プロトコル処理部33は、グローバルアドミッション制御部32によるグローバルアドミッション制御に従い、リソース予約を待っている古いセッションが確立されるまでリソース予約を待っている

新しいセッションに対する予約メッセージの送信を停止する。

【手続補正 2】

【補正対象書類名】明細書

【補正対象項目名】0080

【補正方法】変更

【補正内容】

【0080】ノード装置 3 は、リソース予約を待っている各セッションに対して、セッションが確立されるまでペンディング予約状態を用いて予約メッセージを作成し、作成された予約メッセージを再送する。この場合、受信側ホスト端末装置 4 からみると、最初の予約メッセージが常にリソースの予約に成功し、次の予約メッセージはペンディング予約状態をリフレッシュするためにだけ使用されるように見える。

【手続補正 3】

【補正対象書類名】明細書

【補正対象項目名】0088

【補正方法】変更

【補正内容】

【0088】また、ノード装置 3 は、待機状態にある継続中のセッションのセットアップの順序を決定する際、例えば、q-OSF (q oldest sessions first) 法を用いる。この方法では、選択期間 T_s ごとに、継続中のセッションのグループの中から予約メッセージを伝送する 1 つのセッションがラウンドロビンにより選択される。予約の成功 (予約確認) または予約のキャンセル (予約エラー、予約取り消し、パス取り消しまたはライフタイムの満了) が発生する度に、グループ内のメンバーとなるセッションが更新され、常に q 個の最も古くからリソース予約を待っているセッションからグループが構成される。この場合、再送期間 $T_{round} = q T_s$ となり、継続中のセッションの数 p が q 以上の場合、 $p = q$ に修正される。

【手続補正 4】

【補正対象書類名】明細書

【補正対象項目名】0107

【補正方法】変更

【補正内容】

【0107】次に、図 4 に示すように、ステップ S8 において、プロトコル処理部は、カウント期間 T_c が満了したか否かを判断し、カウント期間 T_c が満了した場合はステップ S9 へ移行し、ネットワークで利用可能な帯域幅 M に含まれる各要求帯域幅 R に対して以降の処理を行い、カウント期間 T_c が満了していない場合はステッ

プ S1 へ戻り、処理を繰り返す。なお、図 4 に示すフローチャートは、ネットワークがサポートする各 R に対して行う。

【手続補正 5】

【補正対象書類名】明細書

【補正対象項目名】0165

【補正方法】変更

【補正内容】

【0165】一方、EP 法および C 法は、大きなネットワークに対しても全く安定である。これは、セッションが確立されるまで、最も古いリソース予約を待っているセッションのリソースをこれらの方法が保持するという特性によるものである。EP 法および C 法のただ 1 つの違いは、C 法は、ネットワークサイズが大きい場合、EP 法よりやや優れている点である。

【手続補正 6】

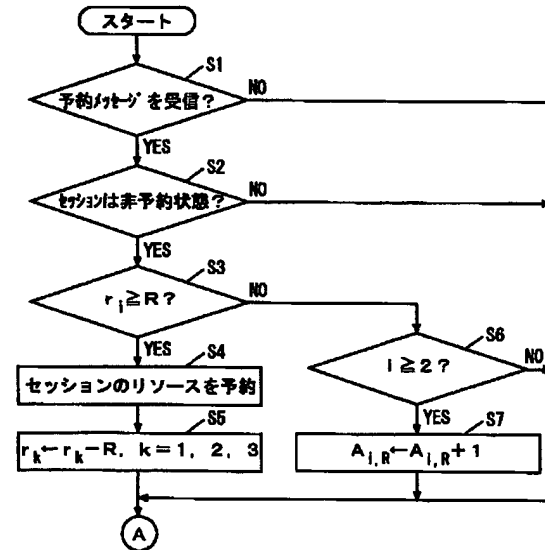
【補正対象書類名】図面

【補正対象項目名】図 3

【補正方法】変更

【補正内容】

【図 3】



【手続補正 7】

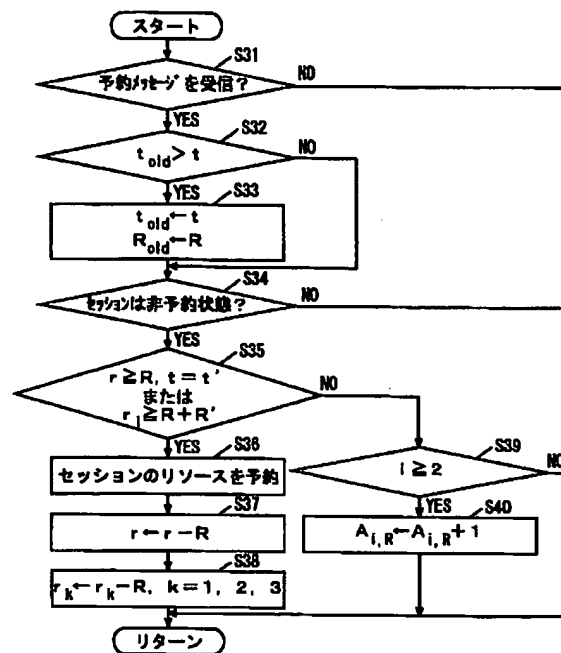
【補正対象書類名】図面

【補正対象項目名】図 6

【補正方法】変更

【補正内容】

【図 6】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.